Fixpoint semantics for proximity-based logic programming

Maximilian Donnermair

2025-05-27

Setting

- Proximity relation $\mathcal{R}$ on an alphabet $\mathcal{L}$ s.t. the *proximity class* of any symbol $\in \mathcal{L}$ is finite
- λ-cut $\in (0, 1]$ and a T-norm $\wedge$
- Logic program P consisting of definite clauses $A \leftarrow B_1, \ldots, B_n$, where $A, B_1, \ldots, B_n$ are atoms
- Semantics $S = \langle D, I \rangle$ with domain D and interpretation function I

Setting

- Proximity relation $\mathcal{R}$ on an alphabet $\mathcal{L}$ s.t. the *proximity class* of any symbol $\in \mathcal{L}$ is finite
- λ-cut $\in (0, 1]$ and a T-norm $\wedge$
- Logic program P consisting of definite clauses $A \leftarrow B_1, \ldots, B_n$, where $A, B_1, \ldots, B_n$ are atoms
- Semantics $S = \langle D, I \rangle$ with domain D and interpretation function I
- $I(f) : D^n \to D$
- $I(p) : D^n \to [0, 1]$
- $S(\wedge) : [0, 1]^2 \to [0, 1]$
- $S(\leftarrow) : [0, 1]^2 \to [0, 1]$

Setting

- Proximity relation $\mathcal{R}$ on an alphabet $\mathcal{L}$ s.t. the *proximity class* of any symbol $\in \mathcal{L}$ is finite
- λ-cut $\in (0,1]$ and a T-norm $\wedge$
- Logic program P consisting of definite clauses $A \leftarrow B_1, \ldots, B_n$, where $A, B_1, \ldots, B_n$ are atoms
- Semantics $S = \langle D, I \rangle$ with domain D and interpretation function I

- variables: $[\![x]\!]^{\sigma}_{S} = \sigma(X)$
- atoms: $[\![p(t_1, \ldots, t_n)]\!]^{\sigma}_{S} = I(p)([\![t_1]\!]^{\sigma}_{S}, \ldots, [\![t_n]\!]^{\sigma}_{S}) \in [0,1]$
- terms: $[\![f(t_1, \ldots, t_n)]\!]^{\sigma}_{S} = I(f)([\![t_1]\!]^{\sigma}_{S}, \ldots, [\![t_n]\!]^{\sigma}_{S}) \in D$
- conjunction of formulas: $[\![F_1 \wedge F_2]\!]^{\sigma}_{S} = \wedge_S([\![F_1]\!]^{\sigma}_{S}, [\![F_2]\!]^{\sigma}_{S})$
- residual of formulas: $[\![F_1 \leftarrow F_2]\!]^{\sigma}_{S} = \leftarrow_S ([\![F_1]\!]^{\sigma}_{S}, [\![F_2]\!]^{\sigma}_{S})$
- all-quantor: $[\![\forall x.F]\!]^{\sigma}_{S} = \forall_S\{[\![F]\!]^{\sigma\{x \to d\}}_{S} \mid d \in D\}$

Setting

- Proximity relation $\mathcal{R}$ on an alphabet $\mathcal{L}$ s.t. the *proximity class* of any symbol $\in \mathcal{L}$ is finite
- λ-cut $\in (0, 1]$ and a T-norm $\wedge$
- Logic program P consisting of definite clauses $A \leftarrow B_1, \ldots, B_n$, where $A, B_1, \ldots, B_n$ are atoms
- Semantics $S = \langle D, I \rangle$ with domain D and interpretation function I

Let B_H be the Herbrand base of our program.

Definition

A fuzzy set $M \in B_H \times (0, 1]$ is a model of P if for every $(A, \alpha) \in M$, $[\![A]\!]_S^\sigma = \alpha \geq \lambda$.

Setting

- Proximity relation $\mathcal{R}$ on an alphabet $\mathcal{L}$ s.t. the *proximity class* of any symbol $\in \mathcal{L}$ is finite
- λ-cut $\in (0, 1]$ and a T-norm $\wedge$
- Logic program P consisting of definite clauses $A \leftarrow B_1, \ldots, B_n$, where $A, B_1, \ldots, B_n$ are atoms
- Semantics $S = \langle D, I \rangle$ with domain D and interpretation function I

We want to compute the *least* or *minimal Herbrand model* M_H (or *canonical model*) of a logic program, i.e. where $\nexists_M \; P \models M \wedge M \subsetneq M_H$. If the model intersection property holds, then $M_H := \{ \bigcap_M \mid P \models M \}$. $\subset$, $\cap$ and $\cup$ (see below) are shall be adequately defined on fuzzy sets.

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \stackrel{?}{\Longrightarrow}$$

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \overset{?}{\Longrightarrow} \text{Canonical Model}$$

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \overset{?}{\Longrightarrow} \text{Canonical Model}$$

Intuitively, that is everything proximal to the program clauses.

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \stackrel{?}{\Longrightarrow} \text{Canonical Model}$$

Intuitively, that is everything proximal to the program clauses.

Example

Let $a \sim_{0.7} b \sim_{0.8} c, q \sim_{0.9} r$, $\lambda = 0.7$, $\wedge = *$ and $P = \{r(a)\}$.
Then deducible are:

- $r(a)$ with degree 1
- $r(b)$ with degree 0.7
- $q(a)$ with degree 0.9

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \stackrel{?}{\Longrightarrow} \text{Canonical Model}$$

Intuitively, that is everything proximal to the program clauses.

Example

Let $a \sim_{0.7} b \sim_{0.8} c, q \sim_{0.9} r$, $\lambda = 0.7$, $\wedge = *$ and $P = \{r(a)\}$.
Not deducible are:

- $q(b)$ because $\mathcal{R}(r(a), q(b)) = 0.63 < 0.7 = \lambda$
- $r(c)$ because $\mathcal{R}(r(a), r(c)) = \mathcal{R}(a, c) = 0$
- $q(c)$ because $\mathcal{R}(r(a), q(c)) = 0$

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \overset{?}{\Longrightarrow} \text{Canonical Model}$$

Intuitively, that is everything proximal to the program clauses.

Example

Let $a \sim_{0.7} b \sim_{0.8} c, q \sim_{0.9} r$, $\lambda = 0.55$, $\wedge = *$ and $P = \{r(b, b)\}$.
Then deducible are:

- $r(c, c)$ with degree 0.64
- $r(a, b)$ with degree 0.7
- $r(a, c)$ with degree 0.56
- $q(a, b)$ with degree 0.63

Proximity and Programs

Given a logic program P, a proximity relation $\mathcal{R}$ with a T-norm $\wedge$ and a λ-cut: Which facts (i.e., elements in the Herbrand base) can we deduce and with which truth degree?

$$\text{Program} \stackrel{?}{\Longrightarrow} \text{Canonical Model}$$

Intuitively, that is everything proximal to the program clauses.

Example

Let $a \sim_{0.7} b \sim_{0.8} c, q \sim_{0.9} r$, $\lambda = 0.55$, $\wedge = *$ and $P = \{r(b, b)\}$.
Not deducible are:

- $q(a, c)$ because $\mathcal{R}(r(b, b), q(a, c)) = 0.504 < 0.55 = \lambda$
- $r(a, a)$ because $\mathcal{R}(r(b, b), r(a, a)) = 0.49 < 0.55 = \lambda$

Proximity and Programs

Explicit notation:

$$
\begin{array}{ll}
r(a): & r(x) \leftarrow x \sim a \\
 & q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
 & q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{array}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.

Proximity and Programs

Explicit notation:

$$
\begin{aligned}
r(a): \qquad & r(x) \leftarrow x \sim a \\
& q(x) \leftarrow q \sim r, x \sim a \\
r(b, b): \qquad & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
& q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{aligned}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We call this procedure *linearization* and *approximation*.

- $lin(p(t_1, \ldots, t_n) \leftarrow B) = p(x_1, \ldots, x_n) \leftarrow x_1 \sim t_1, \ldots, x_n \sim t_n, B$ where the x_i are fresh variables

Proximity and Programs

Explicit notation:

$$
\begin{aligned}
r(a): \quad & r(x) \leftarrow x \sim a \\
& q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): \quad & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
& q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{aligned}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We call this procedure *linearization* and *approximation*.

- $lin(p(t_1, \ldots, t_n) \leftarrow B) = p(x_1, \ldots, x_n) \leftarrow x_1 \sim t_1, \ldots, x_n \sim t_n, B$ where the x_i are fresh variables
- $Lin(P) = \{lin(c) \mid c \in P\}$

Proximity and Programs

Explicit notation:

$$
\begin{aligned}
r(a): \quad & r(x) \leftarrow x \sim a \\
& q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): \quad & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
& q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{aligned}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We call this procedure *linearization* and *approximation*.

- $approx(p(t_1, \ldots, t_n) \leftarrow B) = \{q(t_1, \ldots, t_n) \leftarrow p \sim q, B\}$

Proximity and Programs

Explicit notation:

$$
\begin{aligned}
r(a): \qquad & r(x) \leftarrow x \sim a \\
& q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): \qquad & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
& q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{aligned}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We call this procedure *linearization* and *approximation*.

- $approx(p(t_1, \ldots, t_n) \leftarrow B) = \{q(t_1, \ldots, t_n) \leftarrow p \sim q, B\}$
- $Approx(P) = \bigcup_{c \in P} approx(c)$

Proximity and Programs

Explicit notation:

$$
\begin{array}{ll}
r(a): & r(x) \leftarrow x \sim a \\
 & q(x) \leftarrow q \sim r, x \sim a \\
r(b, b): & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
 & q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{array}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We will need the *ground* instances of these clauses:

- $ground(c) = \{c\theta \mid dom(\theta) = var(c), vran(\theta) = \emptyset\}$

Proximity and Programs

Explicit notation:

$$
\begin{array}{ll}
r(a): & r(x) \leftarrow x \sim a \\
 & q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): & r(x_1, x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
 & q(x_1, x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{array}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a, b)$.
We will need the *ground* instances of these clauses:

- $ground(c) = \{c\theta \mid dom(\theta) = var(c), vran(\theta) = \emptyset\}$
- $Ground(P) = \bigcup_{c \in P} ground(c)$

Proximity and Programs

Explicit notation:

$$
\begin{aligned}
r(a): \qquad & r(x) \leftarrow x \sim a \\
& q(x) \leftarrow q \sim r, x \sim a \\
r(b,b): \qquad & r(x_1,x_2) \leftarrow x_1 \sim b, x_2 \sim b \\
& q(x_1,x_2) \leftarrow q \sim r, x_1 \sim b, x_2 \sim b
\end{aligned}
$$

where $a \sim b$ is interpreted as $\mathcal{R}(a,b)$.

- $Lin(P) = \{lin(c) \mid c \in P\}$
- $Approx(P) = \bigcup_{c \in P} approx(c)$
- $Ground(P) = \bigcup_{c \in P} ground(c)$

Then we call $Ground(Approx(Lin(P)))$ the extended program $\Pi(P)$

Immediate consequence operator

Let H be a Herbrand interpretation and Π an extended program. Then the *immediate consequence operator* is defined as follows:

$$
\begin{aligned}
T_P(H) = \{(A, \alpha) \mid\ & A \leftarrow B_1, \ldots, B_n \in \Pi(P), \\
& \alpha = \beta_1 \wedge \ldots \wedge \beta_n, \\
& (B_1, \beta_1), \ldots, (B_n, \beta_n) \in H, \\
& \} \cup H
\end{aligned}
$$

Immediate consequence operator

Let H be a Herbrand interpretation and Π an extended program. Then the *immediate consequence operator* is defined as follows:

$$\begin{aligned} T_P(H) = \{(A,\alpha) \mid\ & A \leftarrow {B_1}^i, \ldots, {B_{n_i}}^i \in \Pi(P), \\ & \alpha = sup_i\{{\beta_1}^i \wedge \ldots \wedge {\beta_{n_i}}^i\}, \\ & ({B_1}^i, {\beta_1}^i), \ldots, ({B_{n_i}}^i, {\beta_{n_i}}^i) \in H, \\ & 1 \leq i \leq k\} \cup H \end{aligned}$$

Immediate consequence operator

Let H be a Herbrand interpretation and Π an extended program. Then the *immediate consequence operator* is defined as follows:

$$\begin{aligned} T_P(H) = \{(A, \alpha) \mid\ & A \leftarrow {B_1}^i, \ldots, {B_{n_i}}^i \in \Pi(P), \\ & \alpha = sup_i\{{\beta_1}^i \wedge \ldots \wedge {\beta_{n_i}}^i\}, \\ & ({B_1}^i, {\beta_1}^i), \ldots, ({B_{n_i}}^i, {\beta_{n_i}}^i) \in H, \\ & 1 \leq i \leq k\} \cup H \end{aligned}$$

We instantiate the starting Herbrand interpretation H_0 with

$$\{(l \sim l', \mathcal{R}(l, l') \mid l, l' \in \mathcal{L}, \mathcal{R}(l, l') \geq \lambda\}.$$

Immediate consequence operator

Let H be a Herbrand interpretation and Π an extended program. Then the *immediate consequence operator* is defined as follows:

$$\begin{aligned} T_P(H) = \{(A, \alpha) \mid A \leftarrow {B_1}^i, \ldots, {B_{n_i}}^i \in \Pi(P), \\ \alpha = sup_i\{{\beta_1}^i \wedge \ldots \wedge {\beta_{n_i}}^i\}, \\ ({B_1}^i, {\beta_1}^i), \ldots, ({B_{n_i}}^i, {\beta_{n_i}}^i) \in H, \\ 1 \leq i \leq k\} \cup H \end{aligned}$$

The *least fixpoint* of T_P is the smallest H s.t. $T_P(H) = H$. It should coincide with the *least Herbrand model* of P.

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge = \min$ and $P = \{r(a)\}$.

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge = \min$ and $P = \{r(a)\}$. Then $\Pi(P) =$
$\{(r(a) \leftarrow r \sim r, a \sim a), (q(a) \leftarrow q \sim r, a \sim a),$
$(r(b) \leftarrow r \sim r, b \sim a), (q(b) \leftarrow q \sim r, b \sim a),$
$(r(c) \leftarrow r \sim r, c \sim a), (q(c) \leftarrow q \sim r, c \sim a)\}$.

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge =$ min and $P = \{r(a)\}$. Then $\Pi(P) =$
$\{(r(a) \leftarrow r \sim r, a \sim a), (q(a) \leftarrow q \sim r, a \sim a),$
$(r(b) \leftarrow r \sim r, b \sim a), (q(b) \leftarrow q \sim r, b \sim a),$
$(r(c) \leftarrow r \sim r, c \sim a), (q(c) \leftarrow q \sim r, c \sim a)\}$.

$H_0 = \{(a \sim b, 0.7), (b \sim c, 0.5), (r \sim q, 0.8),$
$(a \sim a, 1), (b \sim b, 1), (c \sim c, 1), (r \sim r, 1), (q \sim q, 1)\}$.

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge = \min$ and $P = \{r(a)\}$. Then $\Pi(P) =$
$\{(r(a) \leftarrow r \sim r, a \sim a), (q(a) \leftarrow q \sim r, a \sim a),$
$(r(b) \leftarrow r \sim r, b \sim a), (q(b) \leftarrow q \sim r, b \sim a),$
$(r(c) \leftarrow r \sim r, c \sim a), (q(c) \leftarrow q \sim r, c \sim a)\}$.

$H_1 = T_P(H_0) = H_0 \cup \{$
$(r(a), sup\{1 \wedge 1\}),$
$(q(a), sup\{0.8 \wedge 1\}),$
$(r(b), sup\{1 \wedge 0.7\}),$
$(q(b), sup\{0.8 \wedge 0.7\})$
$\}$

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge = \min$ and $P = \{r(a)\}$. Then $\Pi(P) =$
$\{(r(a) \leftarrow r \sim r, a \sim a), (q(a) \leftarrow q \sim r, a \sim a),$
$(r(b) \leftarrow r \sim r, b \sim a), (q(b) \leftarrow q \sim r, b \sim a),$
$(r(c) \leftarrow r \sim r, c \sim a), (q(c) \leftarrow q \sim r, c \sim a)\}$.

$H_1 = T_P(H_0) = H_0 \cup \{$
$(r(a), 1),$
$(q(a), 0.8),$
$(r(b), 0.7),$
$(q(b), 0.7)$
$\}$

Immediate consequence operator: Example

Let again $a \sim_{0.7} b \sim_{0.5} c, q \sim_{0.8} r$, $\lambda = 0.4$, $\wedge =$ min and $P = \{r(a)\}$. Then $\Pi(P) =$
$\{(r(a) \leftarrow r \sim r, a \sim a), (q(a) \leftarrow q \sim r, a \sim a),$
$(r(b) \leftarrow r \sim r, b \sim a), (q(b) \leftarrow q \sim r, b \sim a),$
$(r(c) \leftarrow r \sim r, c \sim a), (q(c) \leftarrow q \sim r, c \sim a)\}.$

$H_2 = T_P(H_1) = H_1 \cup \emptyset = H_1$
Thus, H_1 is the least fixpoint of T_P.

Immediate consequence operator: Example revisited

Let now $\mathcal{R}' = \mathcal{R}$, but $P' = P \cup \{p(y) \leftarrow r(y)\}$.

- $Lin(P') = Lin(P) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Approx(Lin(P')) = Approx(Lin(P)) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Ground(Approx(Lin(P'))) = \Pi(P') =$
  $Ground(Approx(Lin(P))) \cup \{(p(a) \leftarrow a \sim a, r(a)),$
  $(p(a) \leftarrow a \sim b, r(b)), (p(b) \leftarrow b \sim a, r(a)),$
  $(p(b) \leftarrow b \sim b, r(b)), (p(b) \leftarrow b \sim c, r(c)),$
  $(p(c) \leftarrow c \sim b, r(b)), (p(c) \leftarrow c \sim c, r(c))\}$

Immediate consequence operator: Example revisited

Let now $\mathcal{R}' = \mathcal{R}$, but $P' = P \cup \{p(y) \leftarrow r(y)\}$.

- $Lin(P') = Lin(P) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Approx(Lin(P')) = Approx(Lin(P)) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Ground(Approx(Lin(P'))) = \Pi(P') =$
  $Ground(Approx(Lin(P))) \cup \{(p(a) \leftarrow a \sim a, r(a)),$
  $(p(a) \leftarrow a \sim b, r(b)), (p(b) \leftarrow b \sim a, r(a)),$
  $(p(b) \leftarrow b \sim b, r(b)), (p(b) \leftarrow b \sim c, r(c)),$
  $(p(c) \leftarrow c \sim b, r(b)), (p(c) \leftarrow c \sim c, r(c))\}$

$H_0' = H_0 \cup \{(p \sim p, 1)\}$ and

Immediate consequence operator: Example revisited

Let now $\mathcal{R}' = \mathcal{R}$, but $P' = P \cup \{p(y) \leftarrow r(y)\}$.

- $Lin(P') = Lin(P) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Approx(Lin(P')) = Approx(Lin(P)) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Ground(Approx(Lin(P'))) = \Pi(P') =$
  $Ground(Approx(Lin(P))) \cup \{(p(a) \leftarrow a \sim a, r(a)),$
  $(p(a) \leftarrow a \sim b, r(b)), (p(b) \leftarrow b \sim a, r(a)),$
  $(p(b) \leftarrow b \sim b, r(b)), (p(b) \leftarrow b \sim c, r(c)),$
  $(p(c) \leftarrow c \sim b, r(b)), (p(c) \leftarrow c \sim c, r(c))\}$

$H_0' = H_0 \cup \{(p \sim p, 1)\}$ and
$H_1' = H_0' \cup \{(r(a), 1), (r(b), 0.7), (q(a), 0.8), (q(b), 0.7)\}$.

Immediate consequence operator: Example revisited

Let now $\mathcal{R}' = \mathcal{R}$, but $P' = P \cup \{p(y) \leftarrow r(y)\}$.

- $Lin(P') = Lin(P) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Approx(Lin(P')) = Approx(Lin(P)) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Ground(Approx(Lin(P'))) = \Pi(P') =$
  $Ground(Approx(Lin(P))) \cup \{(p(a) \leftarrow a \sim a, r(a)),$
  $(p(a) \leftarrow a \sim b, r(b)), (p(b) \leftarrow b \sim a, r(a)),$
  $(p(b) \leftarrow b \sim b, r(b)), (p(b) \leftarrow b \sim c, r(c)),$
  $(p(c) \leftarrow c \sim b, r(b)), (p(c) \leftarrow c \sim c, r(c))\}$

$H'_2 = T_P(H'_1) = H'_1 \cup \{$
$(p(a), sup\{1 \wedge 1, 0.7 \wedge 0.7\}),$
$(p(b), sup\{0.7 \wedge 1, 1 \wedge 0.7\}),$
$(p(c), sup\{0.5 \wedge 0.7\})$
$\}$

Immediate consequence operator: Example revisited

Let now $\mathcal{R}' = \mathcal{R}$, but $P' = P \cup \{p(y) \leftarrow r(y)\}$.

- $Lin(P') = Lin(P) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Approx(Lin(P')) = Approx(Lin(P)) \cup \{p(x) \leftarrow x \sim y, r(y)\}$
- $Ground(Approx(Lin(P'))) = \Pi(P') =$
  $Ground(Approx(Lin(P))) \cup \{(p(a) \leftarrow a \sim a, r(a)),$
  $(p(a) \leftarrow a \sim b, r(b)), (p(b) \leftarrow b \sim a, r(a)),$
  $(p(b) \leftarrow b \sim b, r(b)), (p(b) \leftarrow b \sim c, r(c)),$
  $(p(c) \leftarrow c \sim b, r(b)), (p(c) \leftarrow c \sim c, r(c))\}$

$H'_2 = T_P(H'_1) = H'_1 \cup \{$
$(p(a), 1),$
$(p(b), 0.7),$
$(p(c), 0.5)$
$\}$

Proving via SLD-resolution

Definition

Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

Proving via SLD-resolution

Definition

Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

Then we prove $(p(c), 0.5)$ via

$$\leftarrow p(c) \Longrightarrow_{p(y) \leftarrow r(y), y \rightarrow b, 0.5}$$

Proving via SLD-resolution

Definition

Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

Then we prove $(p(c), 0.5)$ via

$$\leftarrow p(c) \Longrightarrow_{p(y) \leftarrow r(y), y \rightarrow b, 0.5}$$
$$\leftarrow r(b) \Longrightarrow_{r(a), id, 1}$$

Proving via SLD-resolution

Definition

Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

Then we prove $(p(c), 0.5)$ via

$$\leftarrow p(c) \Longrightarrow_{p(y) \leftarrow r(y), y \rightarrow b, 0.5}$$

$$\leftarrow r(b) \Longrightarrow_{r(a), id, 1}$$

$$\square$$

Proving via SLD-resolution

Definition

Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

We say that a pair (A, α) is provable from the program P, i.e. $P \vdash (A, \alpha)$, iff $\leftarrow A \Longrightarrow^{*,\alpha}_{WSLD} \quad \square$.

Proving via SLD-resolution

Definition
Let A, A' be atoms and Q, G be conjunctions of atoms. Then

$$\leftarrow A', G \Longrightarrow^{C,\sigma,\beta}_{WSLD} \leftarrow (Q, G)\sigma$$

is a weak SLD-resolution step, where

- $C = A \leftarrow Q \in P$
- $\sigma = wmgu(A, A')$
- $\beta = \mathcal{R}(A\sigma, A'\sigma) \geq \lambda$

We say that a pair (A, α) is provable from the program P, i.e. $P \vdash (A, \alpha)$, iff $\leftarrow A \Longrightarrow^{*,\alpha}_{WSLD} \quad \square$.
$\{(A, \alpha) | P \vdash (A, \alpha)\}$ should again coincide with the least Herbrand model.